



ALGORITHMIQUE

Contrôle Continu n°3

Christophe RAPINE

ENSGI, 1A.

Pour tracer un polygone maya, il nous faut tracer un polygone régulier, en remplaçant chaque côté par une pyramide. Un premier niveau d'analyse donne :

ALGORITHME *polygoneMaya*

ENTRÉES x, y : entiers, n : entier, l : entier

SORTIE

Se placer en (x, y)

Pour $cote := 1$ à n

 Tracer une pyramide de base l à la position courante

 Avancer de l

 Tourner de $360/n$

Fin Pour

Fin *polygoneMaya*

Pour représenter la position courante, le plus simple est de travailler dans le repère d'un Mulot, placé dans le plan et orienté suivant un angle. L'algorithme *pyramide* prendra donc en entrée un Mulot, et les paramètres *base* et *h*. La difficulté est de repositionner le Mulot sur le polygone une fois la pyramide tracée. Une solution est d'utiliser un nouveau Mulot pour chaque pyramide, en l'amenant au préalable au i ème sommet du polygone régulier : c'est à dire en traçant le début d'un polygone.

ALGORITHME *polygoneMaya*

ENTRÉES x, y : entiers, n : entier, l : entier, h : entier

SORTIE

Pour $i := 1$ à n

 Créer un Mulot hector en (x, y)

 // On se place sur le i ème sommet du polygone

 Pour sommet := 1 à $i-1$

 Avancer hector de l

 Faire tourner hector de $360/n$

 Fin Pour

 pyramide(hector, l , h)

Fin Pour

Fin *polygoneMaya*

L'algorithme *pyramide* consiste à itérer le traçage d'un rectangle, en diminuant une dimension de $2h$ à chaque itération, et en se repositionnant. Nous donnons ci-dessus une version possible du programme en Mulot :

Mulot

Proc rectangle(Mulot m , int $coteA$, int $coteB$)

$m.baisse()$;

 Pour $i = 1$ à 2

$m.avance(coteA)$;

$m.tourne(90)$;

$m.avance(coteB)$;

$m.tourne(90)$;

 Ruop

Corp

Proc pyramide(Mulot m , int $base$, int h)

 TantQue ($base > 0$)

 rectangle($m, base, h$) ;

$m.leve()$;

$m.avance(h)$;

$m.tourne(90)$;

$m.avance(h)$;

$m.tourne(-90)$;

$base = base - 2*h$;

 Tnat

$m.tourne(90)$;

Corp

```

Proc polygone(int x, int y, int n, int base, int h)
  angle = 360/n ;
  Pour i = 1 a n
    m = new Mulot(x,y,RED) ;
    Pour j = 1 a i-1
      m.avance(base) ;
      m.tourne(angle);
    Ruop
    Si (i%2 == 0)
    Alors
      m.avance(h);
      pyramide(m,base-2*h,h);
    Sinon
      pyramide(m,base,h);
    Is
  Ruop
Corp

polygone(100,150,8,80,10) ;

Tolum

```